ACSL - Categoria WDTPD (What Does This Program Do)

La această categorie sunt date progrămele scurte (sau secvențe) în limbajul pseudocod și în general trebuie să simulăm executarea lor și să zicem ce rezultat dau pentru anumite valori sau, poate fi și mai complicat, să ne prindem pe cazurile generale ce anume face programul respectiv.

Sintaxa instrucțiunilor în pseudocod este următoarea:

INPUT variable - citire

OUTPUT variabilă - afișare

variable = expression (assignment = atribuire)

IF boolean\_expression THEN - if-ul = instrucțiunea de decizie (branching)

Statement(s) de remarcat că NU are acolade. Pentru a ști unde

ELSE (optional) se termină se folosește END IF

Statement(s)

END IF

WHILE boolean\_expression - repetitiva cu test inițial

Statement(s)

END WHILE

DO - repetitiva cu test final

Statement(s)

WHILE boolean\_expression

care poate avea și forma echivalentă:

REPEAT

Statement(s)

UNTIL not boolean\_expression

FOR variable = start TO end STEP increment - repetitiva cu contor

Statement(s)

NEXT

În cazul în care avem șiruri elementele acestora se scriu de forma A(0) A(1) A(2). Un șir poate începe atât de la 0 cât și de la 1, lucru care se specifică de regulă în enunț. Pentru matrici se scrie la fel, doar că apar două dimensiuni, de exemplu A(0,0) A(0,1), A(1,1),...

În cazul în care avem șiruri de caractere, acestea sunt indexate mereu de la 0. Ele (șirurile de caractere) se scriu între ghilimele:

S="Saguna" - lungimea stringului fiind 6

caracterele fiind S[0]=="S" S[1]=="a" S[2]=="g" S[3]=="u" S[4]=="n" S[5]=="a"

Aici mai pot apărea expresii de genul:

S[:4] = primele 4 caractere din string = "Sagu"

S[4:] = ultimele 4 caractere din string = "guna"

S[2:4] = caracterele dintre indicii specificați = "gun"

Operatorii care pot să apără în pseudocod:  
! (not) , ^ or ↑(exponent), \*, / (**real division**), % (modulus), +, -,

>, <, >=, <=, !=, ==, && (and), || (or) in that order of precedence

Funcții care pot să apară în pseudocod:

abs(x) - absolute value, sqrt(x) - square root, int(x) - greatest integer <= x

Exerciții:







